Homework 11 - Frentes de Pareto

Claudia Lizeth Hernández Ramírez

10 de noviembre de 2021

1. Introducción

Grafica el porcentaje de soluciones de Pareto como función del número de funciones objetivo con diagramas de violín combinados con diagramas de caja-bigote, verificando que diferencias observadas, cuando las haya, sean estadísticamente significativas. Razona en escrito a qué se debe el comportamiento observado.

2. Desarrollo

Inicié trabajando con uno de los códigos vistos en clase [3], agregando dos ciclos `FOR` con los que se variarían las funciones objetivo y se realizarían las iteraciones.

Listing 1: Segmento de código ciclos `FOR` y definición de variables.

```
vc <- 4 #Cuantas variables
md <- 3 #Grado maximo
tc <- 5#Cuantos terminos
funobj = c(2, 3, 4, 5) #Funciones objetivo para k(2, 3, 4, 5)
replicaa = 1:20
for (k in funobj) {
  for (reply in replicaa) {
```

Posteriormente, utilizando como base el códigos que se encuentra en el repositorio de Schaeffer [4] y con ayuda del artículo[5], generé la figura 1.

Listing 2: Segmento de código gráfica.

```
    datos$k = as.factor(datos$k)
    gr = ggplot(datos, aes(x = k, y = porcentaje)) +
      geom_violin(fill = "#C1B3D7", color = "#A589C1")
    gr + geom_boxplot(width = 0.1, fill = "#A5DEEE",
                      color = "black", lwd = 0.3)+
      theme(panel.background = element_rect(fill = "#FDDEEE",
                                            color = "black")) +
      labs(x = "N\'umero de funciones objetivo",
           y = "Porcentaje de soluciones Pareto")
```

Como es costumbre, realicé pruebas estadísticas a mis resultados, los resultados se encuentran en la sección de `Estadística`.

Listing 3: Segmento de código pruebas estadísticas.

```
#Estad\'isitica prueba de normalidad -
    #con p menor a 0.05 se rechaza hipotesis nula H0
    #H0: los datos proceden de una distribuci\'on normal
    #H1: los datos no proceden de una distribuci\'on normal
tapply(datos$porcentaje, datos$k, shapiro.test)

#PRUEBA ESTADISTICA
datos%>%
  group_by(k) %>%
  summarise(
    cantidad_de_participantes = n(),
    promedio = mean(porcentaje, na.rm = TRUE),
    desviacion_estandar = sd(porcentaje, na.rm = TRUE),
    varianza = sd(porcentaje, na.rm = TRUE)^2,
    mediana = median(porcentaje, na.rm = TRUE),
    rango_intercuartil = IQR(porcentaje, na.rm = TRUE)
  )

kruskal.test(porcentaje ~ k, data = datos)
pairwise.wilcox.test(datos$porcentaje, datos$k)
```

3. Estadística

Cuadro 1: Resultados obtenidos de prueba de normalidad de Shapiro.

K	W value	P value	¿Se acepta H0?
2	0.7083	$4{,}93 \times 10^{-5}$	no
3	0.6697	$1{,}70 \times 10^{-5}$	no
4	0.9580	0.5052	sí
5	0.9097	0.0631	sí

Cuadro 2: Resultados obtenidos de prueba Kruskal-Wallis.

Chi cuadrada	DF	P
47.786	3	$2{,}365 \times 10^{-10}$

Cuadro 3: Diferencias entre grupos. Paiwise Wilcox.

	2	3	4
3	0.00035		
4	$1{,}2 \times 10^{-6}$	0.00324	
5	$5{,}3 \times 10^{-7}$	0.00035	0.33680

Cuadro 4: Información individual de los datos.

Carga	Qty. Participantes	promedio	Desv. Std.	Varianza	Mediana	Rango Intercuartil
2	20	2.72	2.84	8.07	1.75	2.0
3	20	17.00	22.70	515.00	6.75	14.4
4	20	37.80	22.30	496.00	39	34.1
5	20	48.80	29.80	887.00	42.2	56.0

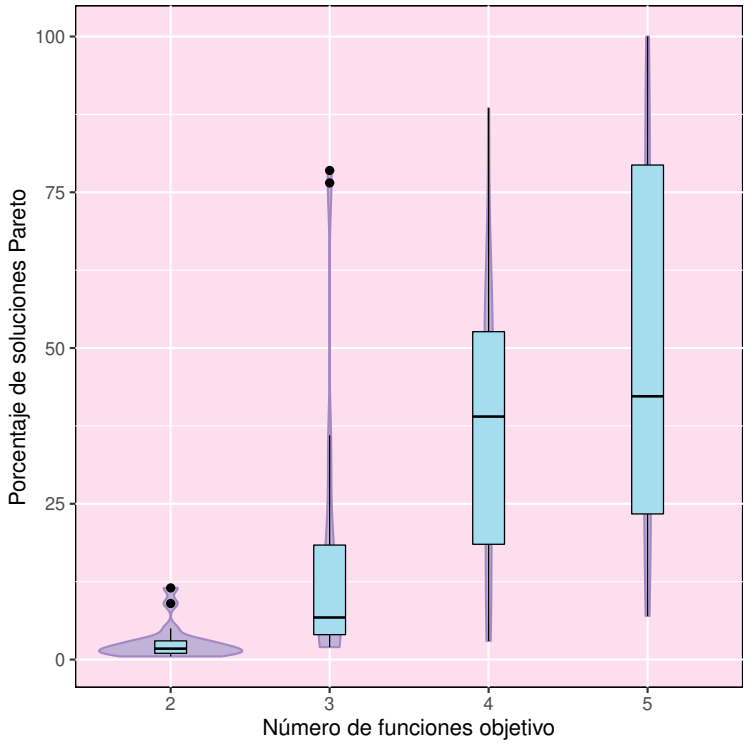


Figura 1: Porcentaje de soluciones de Pareto vs Función objetivo.

4. Conclusión

Como se puede observar en la figura 1, el aumento en el porcentaje de soluciones de Pareto, depende fuertemente del número de funciones objetivo que se tengan.
Entre menos funciones objetivo se tengan, el porcentaje de Pareto será menor; Por otro lado, mientras mas funciones objetivo se tengan, el porcentaje de Pareto aumenta hasta lograr una optimización de casi el 100 %. Se puede esta relación se puede explicar en base a que, como se tiene una mayor cantidad de funciones para optimizar, si bien no estarán optimizadas en todos los criterios, lo estarán para algunos otros, satisfaciendo así a por lo menos alguna de las otras funciones.
Esto se comprueba con las pruebas estadísticas que se aplicaron, ya que con una `P =` $2{,}365 \times 10^{-10}$ se rechaza la hipótesis nula, lo cual nos indica que existen diferencias significativas entre las funciones objetivo.

5. Reto 1

Para entender de que trataba el reto 1, me basé en un video de semestres anteriores [6], asi mismo como en el repositorio de una compañera [1] y en un sitio web para entender que significaba `k-means` [2].

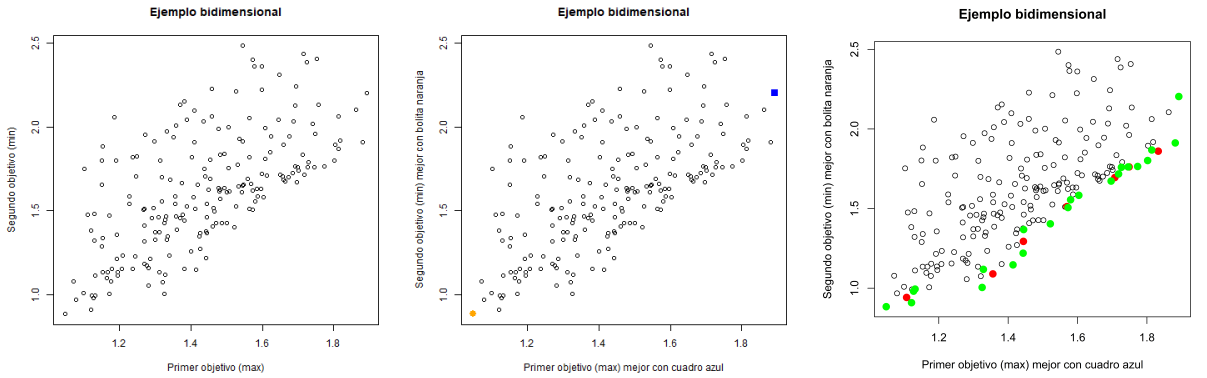
Listing 4: Segmento de código reto 1.

```
frente <- subset(val, no.dom) # solamente las no dominadas
    print(frente)
```

```
data = rbind(data, frente)
names(data) = c("X", "Y")
distance <- dist(data, method = "euclidean")
pp = 0.4
res.k = kmeans(data, centers = (dim(frente)[1]*pp), algorithm = "Lloyd")
res.k$cluster
res.k$centers
dete = rbind(dete, res.k$centers)

    png("p11_frentesub.png", width=15, height=15, units="cm", res=1200)
    plot(val[,1], val[,2], xlab=paste(xl, "mejor con cuadro azul"),
         ylab=paste(yl,"mejor con bolita naranja"),
         main="Ejemplo bidimensional")
    points((dete[,1]), (dete[,2]), col="red", pch=16, cex=1.5)
    points((frente[,1]), (frente[,2]), col="green", pch=16, cex=1.5)
    mejor1 <- which.max((1 + (-2 * minim[1])) * val[,1])
    mejor2 <- which.max((1 + (-2 * minim[2])) * val[,2])
    graphics.off()
```

Figura 2: Frente de Pareto reto 1.



(a) inicial (b) Mejores (c) Frentes diversificados con 40 %

Referencias

[1] Elisa Schaeffer. Repositorio fabiola, 2020. URL https://github.com/fvzqa/Simulacion/tree/master/Tarea11.

[2] Elisa Schaeffer. Teoría k-means, 2020. URL https://statsandr.com/blog/clustering-analysis-k-means-and-hierarchical-clustering-by-hand-and-in-r/.

[3] Elisa Schaeffer. Codigo frentes de pareto, 2021. URL https://github.com/satuelisa/Simulation/blob/master/ParetoFronts/front.R.

[4] Elisa Schaeffer. Codigo clase, 2021. URL https://github.com/satuelisa/Simulation/blob/master/ParetoFronts/violin.R.

[5] Elisa Schaeffer. Estetica ggplot2, 2021. URL http://www.sthda.com/english/wiki/ggplot2-themes-and-background-colors-the-3-elements.

[6] Elisa Schaeffer. Video reto 1, 2021. URL https://youtu.be/Blek9NZsbU4.